E 8957
11-10-94

NASA Technical Memorandum 106649

Graphical User Interface for the NASA FLOPS Aircraft Performance and Sizing Code

Thomas M. Lavelle and Brian P. Curlett
Lewis Research Center
Cleveland, Ohio

October 1994

National Aeronautics and
Space Administration

GRAPHICAL USER INTERFACE FOR THE NASA FLOPS AIRCRAFT PERFORMANCE AND SIZING CODE

Thomas M. Lavelle and Brian P. Curlett
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

SUMMARY

XFLOPS is an X-Windows/Motif graphical user interface for the aircraft performance and sizing code FLOPS. This new interface simplifies entering data and analyzing results, thereby reducing analysis time and errors. Data entry is simpler because input windows are used for each of the FLOPS namelists. These windows contain fields to input the variable's values along with help information describing the variable's function. Analyzing results is simpler because output data are displayed rapidly. This is accomplished in two ways. First, because the output file has been indexed, users can view particular sections with the click of a mouse button. Second, because menu picks have been created, users can plot engine and aircraft performance data. In addition, XFLOPS has a built-in help system and complete on-line documentation for FLOPS.

INTRODUCTION

The Flight Optimization System (FLOPS) (ref. 1) is a multidisciplinary computer program for designing and evaluating advanced aircraft concepts. It consists of nine primary analysis modules: (1) weights, (2) aerodynamics, (3) engine cycle analysis, (4) propulsion data scaling and interpretation, (5) mission performance, (6) takeoff and landing, (7) noise footprint, (8) cost analysis, and (9) program control. Together these modules predict the aircraft's overall performance, weight, cost, and environmental factors–such as noise and emissions–needed for advanced concept evaluations.

Obviously, this is a very complex analysis, requiring many different input parameters and output data for all of the different disciplines. In fact, FLOPS has over 1000 possible input parameters and several modes of operation. It also has many types of output, pertaining to the different aspects of aircraft performance. Keeping track of all this information can be a difficult task, even for experienced FLOPS users.

XFLOPS was created to make data entry and output analysis easier. These are simplified by XFLOPS' graphical environment, which makes assistance available at the touch of a button. To make data entry easier, an input form is provided for each namelist. These forms contain fields for all the data in a namelist. Each field contains the variable name, its current value, and a short description of its function. Complete help information and other functions are available at the touch of a button. In addition, comment blocks are inserted between groups of variables to describe how they work together. These forms, referred to as "namelist windows," serve as a working user's guide. They eliminate the need for hard copy documentation, which tends to become outdated. A typical XFLOPS screen is shown in figure 1. The namelist window is in the lower right corner of the screen.

FLOPS requires some data to be entered in fixed-field formats. XFLOPS makes entering this type of data easier because users place these values in an array field editor. This editor eliminates the worry about positioning data in exact columns.

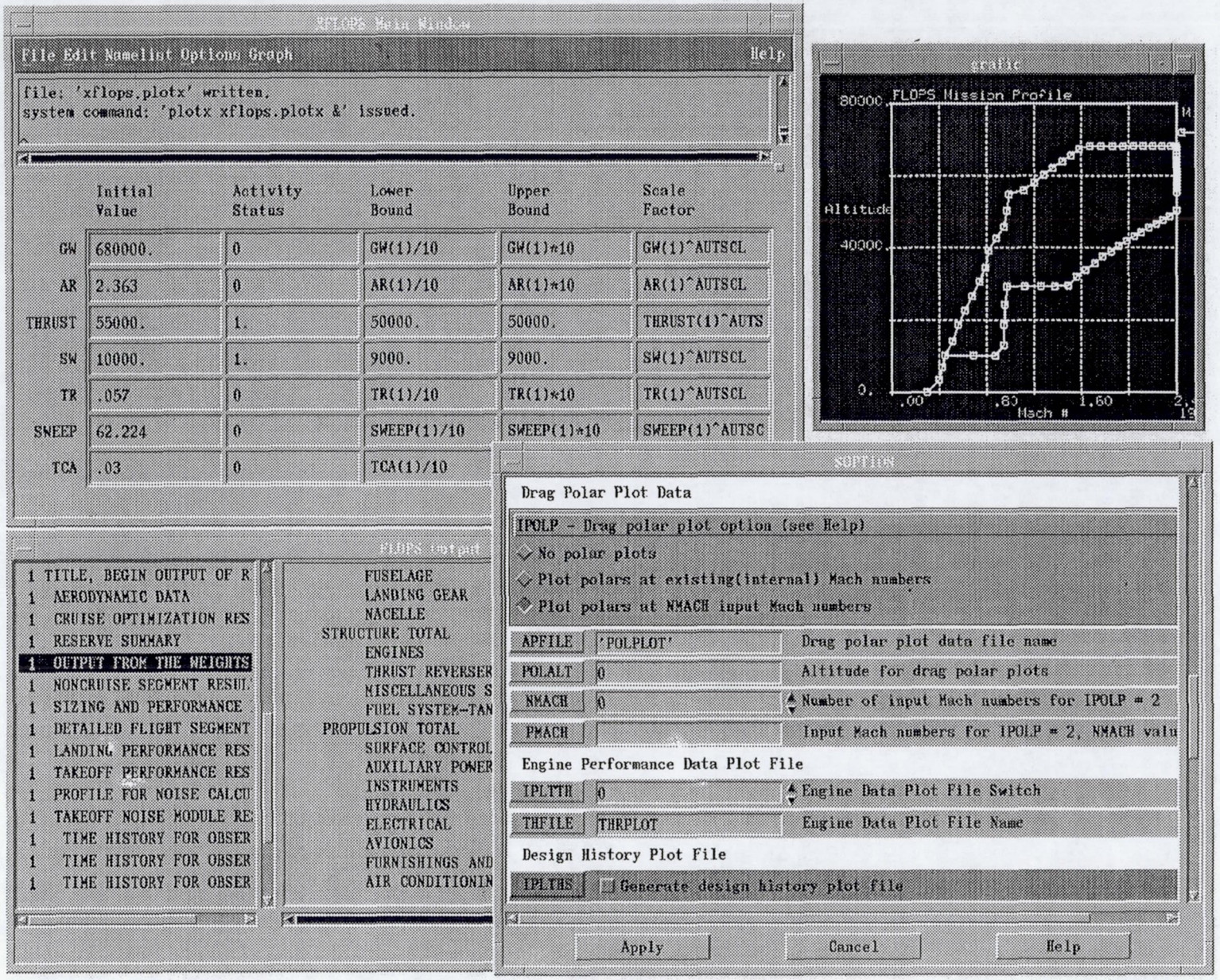


Figure 1: XFLOPS Windows.

Several functions were added to make analyzing results easier. The output file is simpler to read because it is divided into manageable sections. Individual sections can quickly be displayed through a table of contents. In addition, predefined plot functions allow the user to quickly obtain plots of different aspects of aircraft performance, making trends and mistakes much easier to find than they are in a table of numbers.

XFLOPS works as a preprocessor and a post-processor. It reads in a standard FLOPS input file, allows parameters to be modified, creates a new input file, and then executes FLOPS independently. Output operations use standard FLOPS output files. Thus, there are no direct ties between XFLOPS and FLOPS. XFLOPS can be used to examine preexisting input and output files.

The user interface code, NLEdit, serves as the backbone for XFLOPS (ref. 2). NLEdit is a framework for creating graphical user interfaces for FORTRAN programs that rely on namelist input. Unlike other user interface development tools, NLEdit only requires the programmer to describe the data, not the actual interface. An appropriate interface is automatically generated from a standard ASCII file in which the data are described. This file, referred to as the "data definition file," is the key to XFLOPS. Any computer code that addresses complex changing problems, like FLOPS, must be continually modified and updated or otherwise become obsolete. With XFLOPS, the only user interface programming required is for updating the data definition file. The user interface automatically updates itself to reflect these changes.

This report describes how XFLOPS was created with the NLEdit user interface framework. Most of the NLEdit routines were used without modification. New code was written to handle fixed-field inputs and post-processing, which are not handled by NLEdit. Appendix A contains more information on FLOPS, and appendixes B and C contain the on-line help files for XFLOPS.

HARDWARE/SOFTWARE COMPATIBILITY

XFLOPS is written in ANSI standard C using the OSF/Motif widget set. Motif is the industry standard toolkit for the programming of graphical user interfaces on UNIX workstations (refs. 3 and 4).

Although XFLOPS was developed on an IBM RS/6000 workstation, it has been ported to and tested on Silicon Graphics, Inc., IRIS workstations and Sun Microsystems Corp. Sun workstations. FLOPS/XFLOPS should port to any UNIX computer that has ANSI C and FORTRAN 77 compilers, X11R4 and OSF/Motif 1.1 or higher. We recommend using a computer with at least 32MB of RAM.

INPUT DATA

Input data are divided into two sections: namelist input and fixed-field input. Most of the input data are namelist input and were handled with NLEdit. The fixed-field input data (engine, aerodynamics, and mission data) were handled by using a matrix editor and special code written for this application.

Namelist Input

All user interface requirements for FLOPS namelist input were handled by using NLEdit without any modifications. Thus, XFLOPS has all of NLEdit's functionality. This includes the

```
  /*Input field description for comment line.*/
option_msg5 C 1 1 ! ! !
 msg
 Drag Polar Plot Data
.
  /*Input field description for multiple choice variable IPOLP.  It has*/
  /*three possible values; 0, 1, and 2, and a default value of 0.      */

IPOLP M 3 1   0  0,1,2, "No polar plots",\
"Plot polars at existing(internal) Mach numbers",\
"Plot polars at NMACH input Mach numbers"
 Drag polar plot option (see help)
 Drag polar plot option
 = 0, No drag polar plots (Default)
 = 1, Prepare data on APFILE (Unit 8) for plotting
      drag polars at existing (internal) Mach numbers
 = 2, Drag polar plot data will be at NMACH input
      Mach numbers
.
  /*Input field description for filename variable APFILE.  It has a*/
  /*default value of 'POLPLOT'.                                      */
APFILE F 1 1  'POLPLOT' ! !
 Drag polar plot data file name
 Drag polar plot data file name
 (Default = 'POLPLOT')
.
  /*Input field description for real variable POLALT.  It has a default*/
  /*value of 0 and no minimum and maximum.                             */

POLALT f 1 1  0 ! !
 Altitude for drag polar plots
 Altitude for drag polar plots
.
  /*Input field description for integer variable NMACH.  It has a      */
  /*default value of 0, a minimum value of 0, and a maximum value of 20*/
NMACH i  1 1   0 0 20
 Number of input Mach numbers for IPOLP = 2
 Number of input Mach numbers for IPOLP = 2
.
  /*Input field description for real array PMACH.  It is a 20 by 1 */
  /*array with no default, minimum, or maximum.                    */
PMACH f  20 1 ! ! !
 Input Mach numbers for IPOLP = 2, NMACH values
 Input Mach numbers for IPOLP = 2, NMACH values
.
```

Figure 2: Section of XFLOPS data definition file.

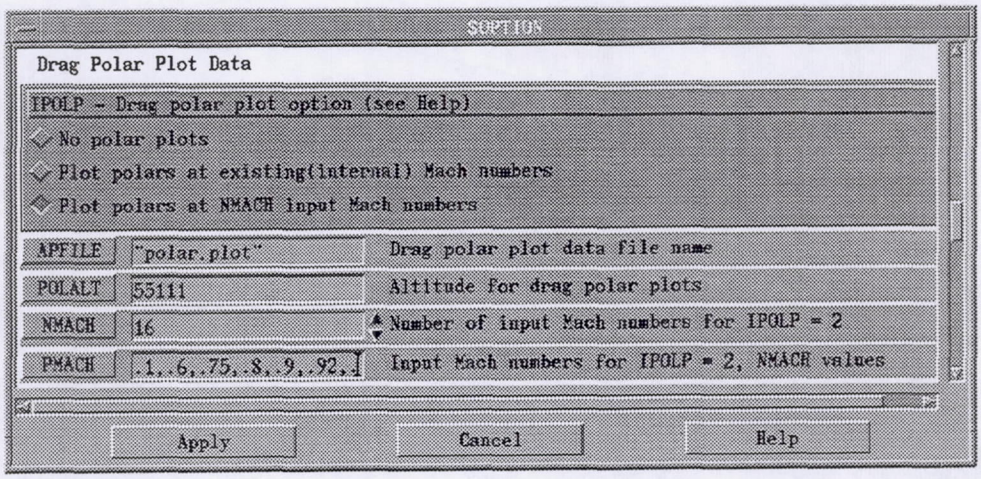


Figure 3: Graphical user interface resulting from data definition file shown in figure 2.

on-line editor, on-line help, variable search capability, show defaults option, show only used parameters option, and test limits option. Actually, many of these functions were created because of XFLOPS requirements (they are now part of NLEdit).

The first step in creating XFLOPS was to create a data definition file with information about the FLOPS input parameters. Each input parameter was described according to type (e.g., integer, real, or Boolean), dimensions, default value, minimum value, maximum value, label line, and help information. Figure 2 shows a portion of the XFLOPS data definition file, and figure 3 shows the resulting namelist input window.

The parsing routine for the data definition file uses a simple format. All data types use the same format although some of the fields may have a slightly different meaning based on variable type. The format of each data item is as follows:

```
most inputs:

name type m n default   min   max
 one line label string.
 all the help information you want to
 type on as many lines as you like.
.

mutiple choice inputs:

name type m n default  possible inputs   description of possible inputs
 one line label string.
 all the help information you want to
 type on as many lines as you like.
.
```

The first field of a parameter's description, "name," is the variable name. The second field is a flag indicating the variable type. The "type" field can be one of the following:

i integer

f floating point

d double precision floating point

s character string

F filename

C comment

B Boolean (true or false)

M multiple choice

The third and fourth fields, "m" and "n," are array dimensions. The fifth field, "default," is the default value. For numeric variables, the sixth and seventh values, "min" and "max," are the minimum and maximum values. For multiple-choice inputs, the sixth and seventh values, "possible inputs" and "description of possible inputs," are the possible selections and descriptions of these possible selections. The first line below the input is the label line to be printed on screen. The rest of the input field is additional help information available from a menu selection. For more information on the options available and the format of the data definition file, see the NLEdit reference (ref. 2) and the XFLOPS help file (app. B).

The data definition file is read in when XFLOPS is started. To add new variables to the interface, users need only edit the data definition file. No other programming is necessary.

One important NLEdit feature that was added, with FLOPS in mind, is the search function. This function searches the variable name, label, and/or help information for a user-entered string. It then displays in a window all the matching variables found in the database. This feature is extremely helpful, especially for new users. Suppose a user wants to list all the input data that pertain to the aircraft nacelle. By searching on the string "nacelle," the user can display all the nacelle-related input data in one form, regardless of their namelist location. The search window is shown in figure 4, and the search result window is shown in figure 5.

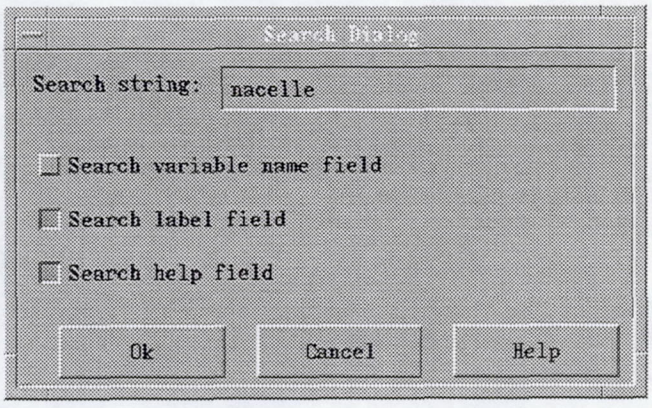


Figure 4: Search prompt window for nacelle-related input data.

In addition to the namelist input windows, a special input field was created in the main window for the often-changed, overall design variables: gross weight, aspect ratio, thrust, wing size, taper ratio, sweep, and thickness-to-chord ratio. This window allows users to edit the design variable values, lower bounds, upper bounds, and scale factor without opening a namelist window (these variables are from the CONFIN namelist). The main window is shown in figure 6.

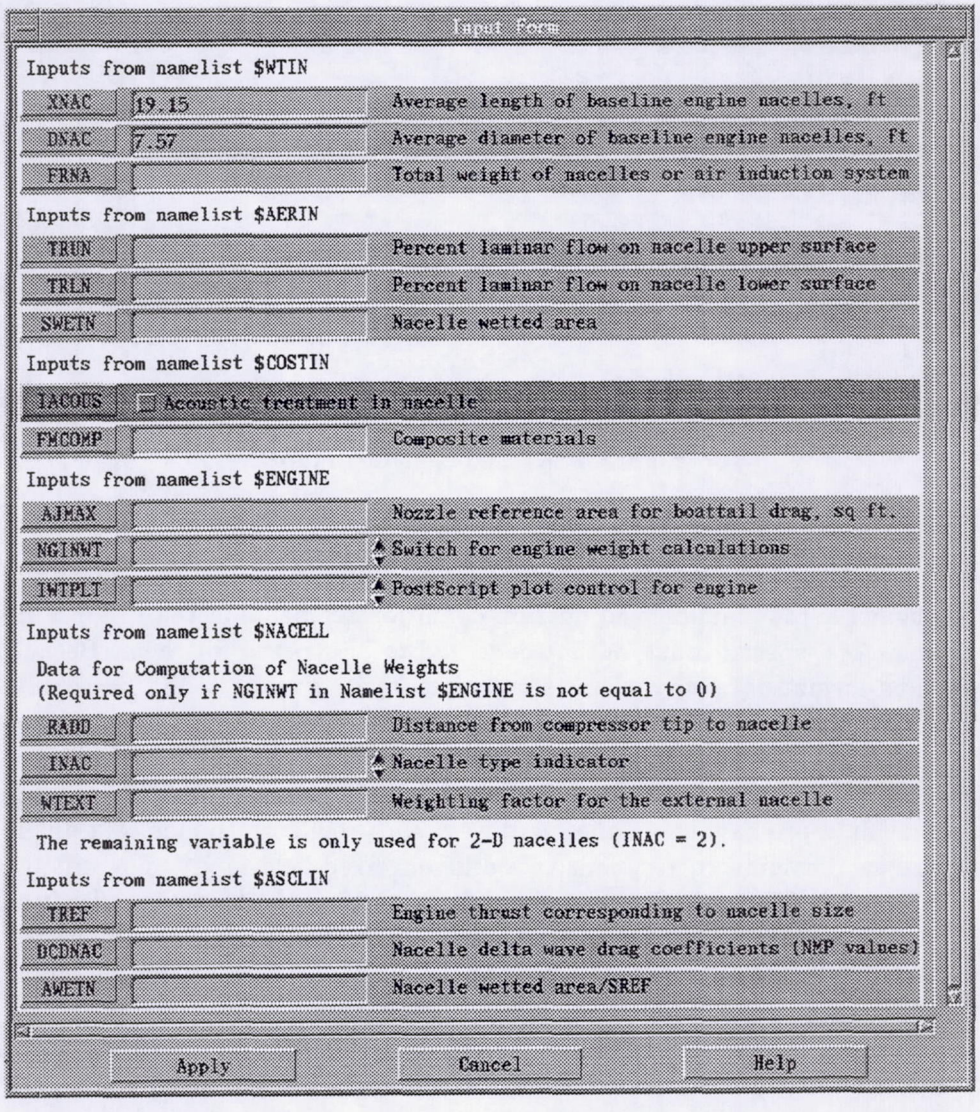


Figure 5: Search result showing all the nacelle-related input data.

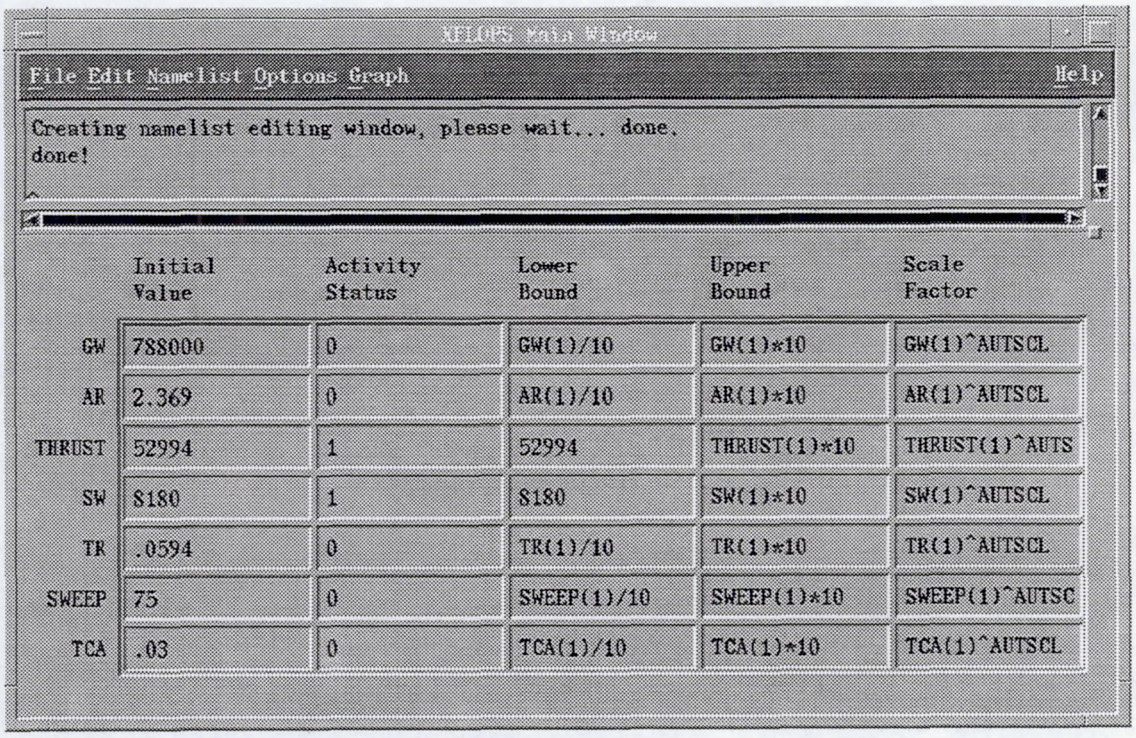


Figure 6: XFLOPS main window.

Fixed-Format Input Data

The engine, aerodynamics, and mission definition segments of the FLOPS input are all formatted. Therefore, special input/output code had to be written for them. Because the locations of fixed-format data vary, there is no way to write generic code. These data are all treated as two-dimensional arrays. Therefore, the matrix feature in NLEdit was used to edit them. The engine data are treated as a matrix: each row corresponds to an individual data point, and each column corresponds to a different type of data (e.g., Mach number, altitude, etc.). The mission segment data are handled in the same way: each row corresponds to a different segment card input, and each column corresponds to a different fixed-field input. The different types of aerodynamic data are also read in as two-dimensional arrays: the top row and left column are the independent variables (Mach number, altitude, and lift coefficient (CL)), and the rest of the array comprises the dependent drag data. A sample of the original FLOPS aerodynamic input and the corresponding XFLOPS input form are shown in figures 7 and 8, respectively.

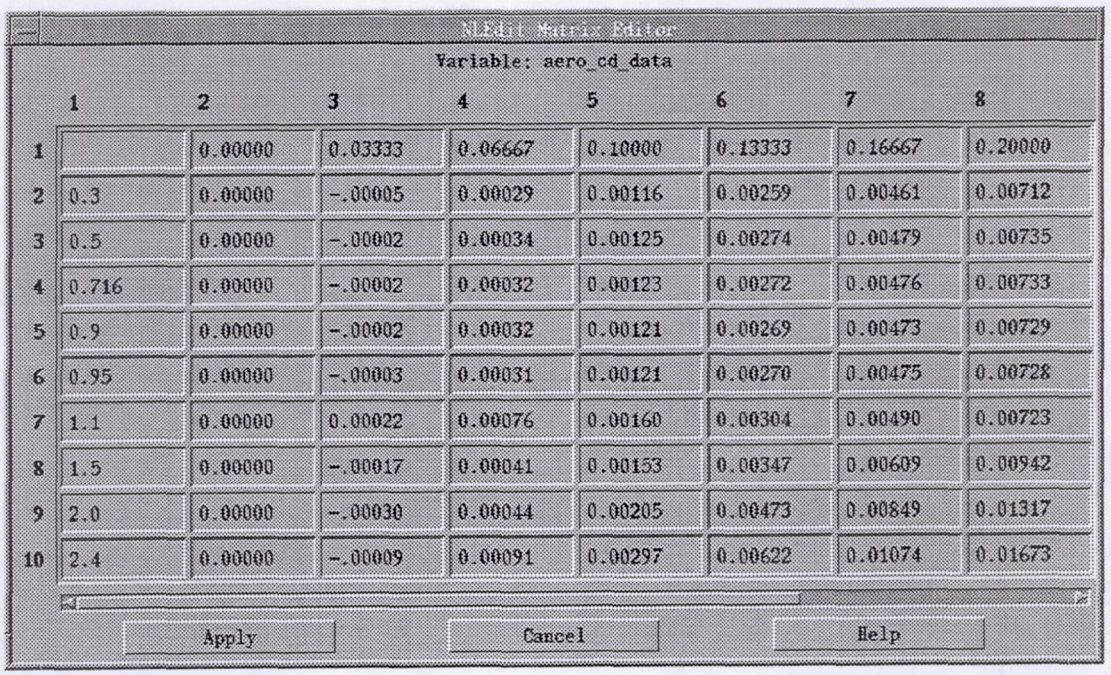
The advantage of the XFLOPS input forms should be obvious. When using the old method of inputting data, a user had to count rows and columns to be sure the data were in exactly the right location. With the new method, a user needs only to find the correct input cell.

RUNNING FLOPS

Once a user has entered all the data, XFLOPS can generate an input file and execute FLOPS. When creating a FLOPS input file, XFLOPS will only print out the values read in from the original file or entered from the interface. It will not print out every FLOPS variable. This approach makes the file more readable and ensures that the correct FLOPS default values are being used.

```
        9  10 (Array size: 9 Mach numbers and 10 CL's)
(Mach Nos.) 0.3     0.5    0.716     0.9     0.95      1.1      1.5      2.0
2.4
 (CL's)  0.00000 0.03333 0.06667 0.10000 0.13333 0.16667 0.20000 0.23333 0.26667 0.30000
 -----   0.00000 -.00005 0.00029 0.00116 0.00259 0.00461 0.00712 0.01027 0.01401 0.01829
   |     0.00000 -.00002 0.00034 0.00125 0.00274 0.00479 0.00735 0.01069 0.01478 0.01956
   |     0.00000 -.00002 0.00032 0.00123 0.00272 0.00476 0.00733 0.01076 0.01506 0.02016
   |     0.00000 -.00002 0.00032 0.00121 0.00269 0.00473 0.00729 0.01075 0.01515 0.02034
 (CD's)  0.00000 -.00003 0.00031 0.00121 0.00270 0.00475 0.00728 0.01075 0.01514 0.02034
   |     0.00000 0.00022 0.00076 0.00160 0.00304 0.00490 0.00723 0.01005 0.01340 0.01744
   |     0.00000 -.00017 0.00041 0.00153 0.00347 0.00609 0.00942 0.01385 0.01877 0.02369
   |     0.00000 -.00030 0.00044 0.00205 0.00473 0.00849 0.01317 0.01939 0.02621 0.03279
 -----   0.00000 -.00009 0.00091 0.00297 0.00622 0.01074 0.01673 0.02292 0.02992 0.03652
```

Figure 7: FLOPS drag coefficient (CD) input section.



	1	2	3	4	5	6	7	8
1		0.00000	0.03333	0.06667	0.10000	0.13333	0.16667	0.20000
2	0.3	0.00000	-.00005	0.00029	0.00116	0.00259	0.00461	0.00712
3	0.5	0.00000	-.00002	0.00034	0.00125	0.00274	0.00479	0.00735
4	0.716	0.00000	-.00002	0.00032	0.00123	0.00272	0.00476	0.00733
5	0.9	0.00000	-.00002	0.00032	0.00121	0.00269	0.00473	0.00729
6	0.95	0.00000	-.00003	0.00031	0.00121	0.00270	0.00475	0.00728
7	1.1	0.00000	0.00022	0.00076	0.00160	0.00304	0.00490	0.00723
8	1.5	0.00000	-.00017	0.00041	0.00153	0.00347	0.00609	0.00942
9	2.0	0.00000	-.00030	0.00044	0.00205	0.00473	0.00849	0.01317
10	2.4	0.00000	-.00009	0.00091	0.00297	0.00622	0.01074	0.01673

Figure 8: XFLOPS drag coefficient (CD) data window.

Once the input file has been created, the standard FLOPS code is executed. Because XFLOPS works with a standard input file, it can be used to examine preexisting input files, create input files for use at a later date or on another system, or for full FLOPS execution. After FLOPS is executed, XFLOPS reads in data from standard FLOPS output files. Thus, XFLOPS can examine data that were not generated by the interface.

OUTPUT

XFLOPS has several special functions designed to make analyzing results easier. These include an indexed output viewing window, numerous plotting functions, and a separate graphical user interface (XFLOPSPLOT) for thumbprint plotting. The resulting ease of processing and viewing voluminous output data simplifies the process of interpreting results and identifying errors.

Viewing Output Files

The output viewing window is shown in figure 9. The left side of the window is a table of contents. FLOPS flags section breaks by placing the character # in the first column of the output file. The table of contents is generated from these section breaks. When a user clicks on a section title, that section of the output file is displayed in the window to the right. The numbers in the first column of the table of contents indicate the occurrence numbers for those types of data in the output file. (2 RESERVE SUMMARY indicates the second occurrence of a RESERVE SUMMARY block in the output file.) This editor simplifies navigating through large output files.

XFLOPS also provides a text-editing window with menu selections to quickly load the various input/output files used by FLOPS. This mouse-driven window provides most standard editing services.

X-Y Plots

Line plots are created with a stand-alone plotting program called Plotx. This program reads a namelist input file that indicates the location of the data to be plotted–file name, column number, number of points, or other data–and plot description information–titles, symbols and line style. Plotx reads the plot data file and calls a Grafic routine to display the plot on the screen. Grafic, a set of FORTRAN callable routines for plotting data, is public domain software available free of charge.[1] It provides many useful interactive features, including annotation, zooming, display of a point's coordinates, and postscript output file creation. For more information, see the Grafic help available with XFLOPS.

Users can plot aerodynamics, engine, noise, overall mission, takeoff, climb, cruise, and descent data. There are many types of plots available from within XFLOPS. When a plot is chosen, XFLOPS will search for the appropriate FLOPS file. It will then create a Plotx input file and issue a system command to execute Plotx. Some plots may have more than one set of data. For example, there may be several cruise schedule outputs in one FLOPS run. In this case, all the schedules are plotted on one graph. Several sample plots are shown in figures 10 to 13.

Adding additional plot functions is easy, requiring a routine that does only three things. First, it must store the Plotx titles in the database. Second, it must store the plot data location in the database. Finally, it must execute Plotx.

[1] Available via anonymous ftp from swedishchef.lerc.nasa.gov

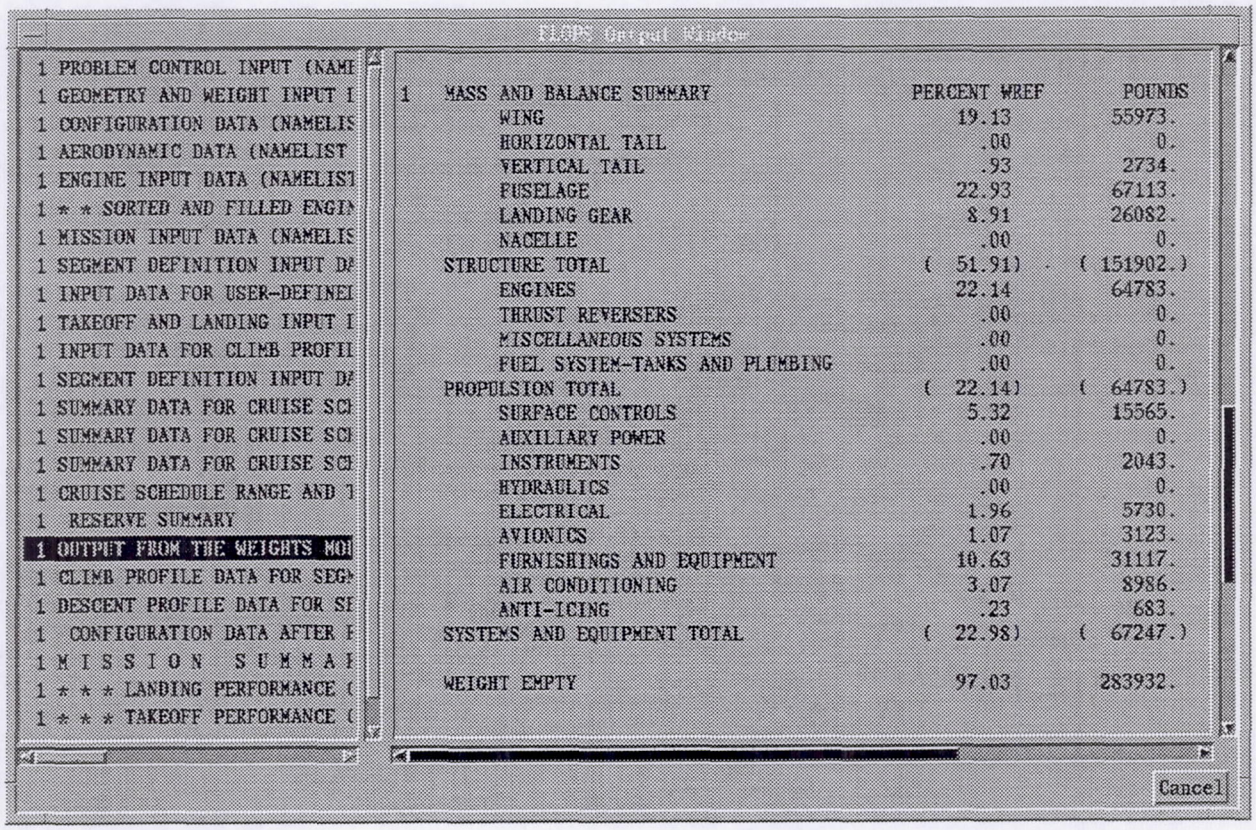


Figure 9: XFLOPS segmented output window.

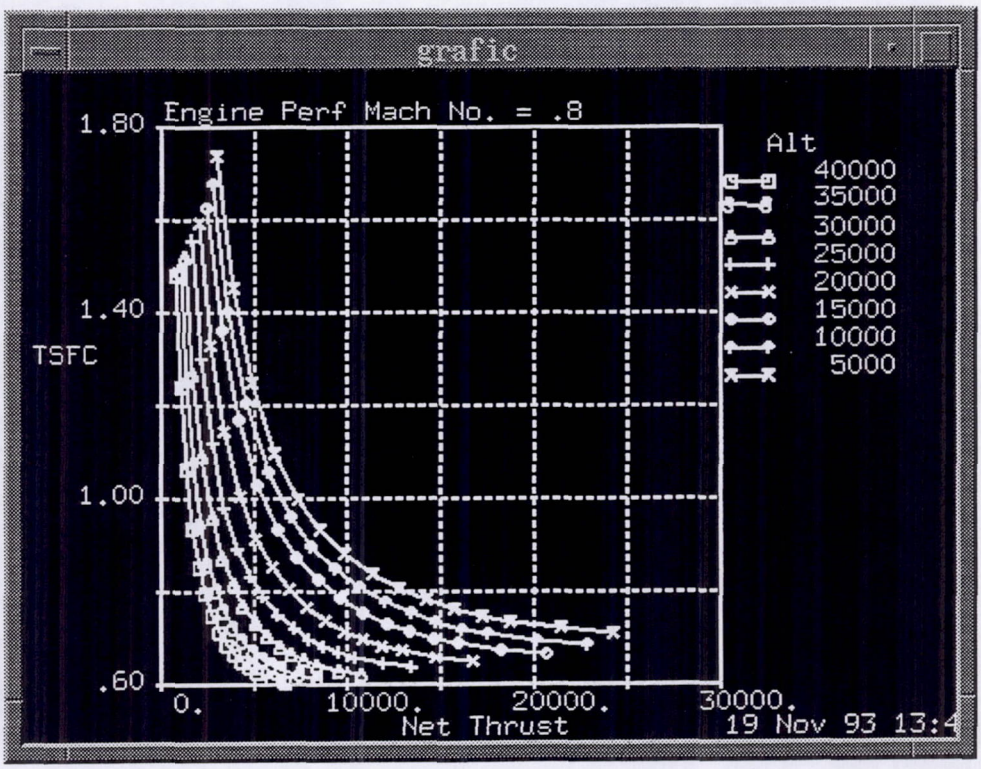


Figure 10: XFLOPS engine throttle curve.

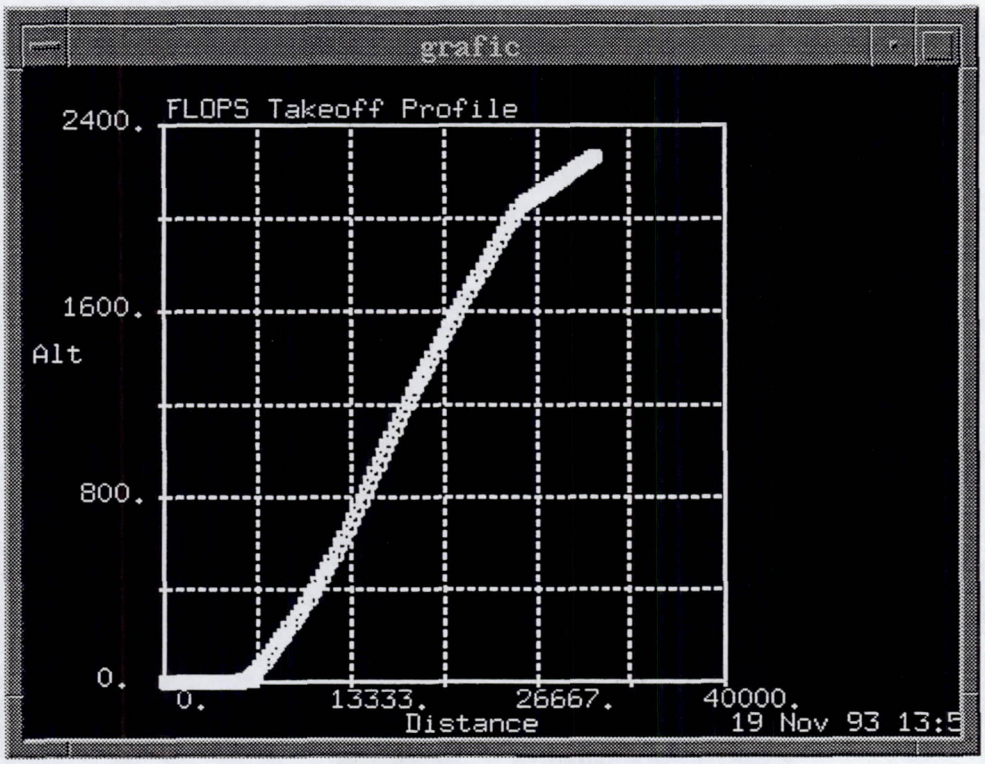


Figure 11: XFLOPS takeoff plot.

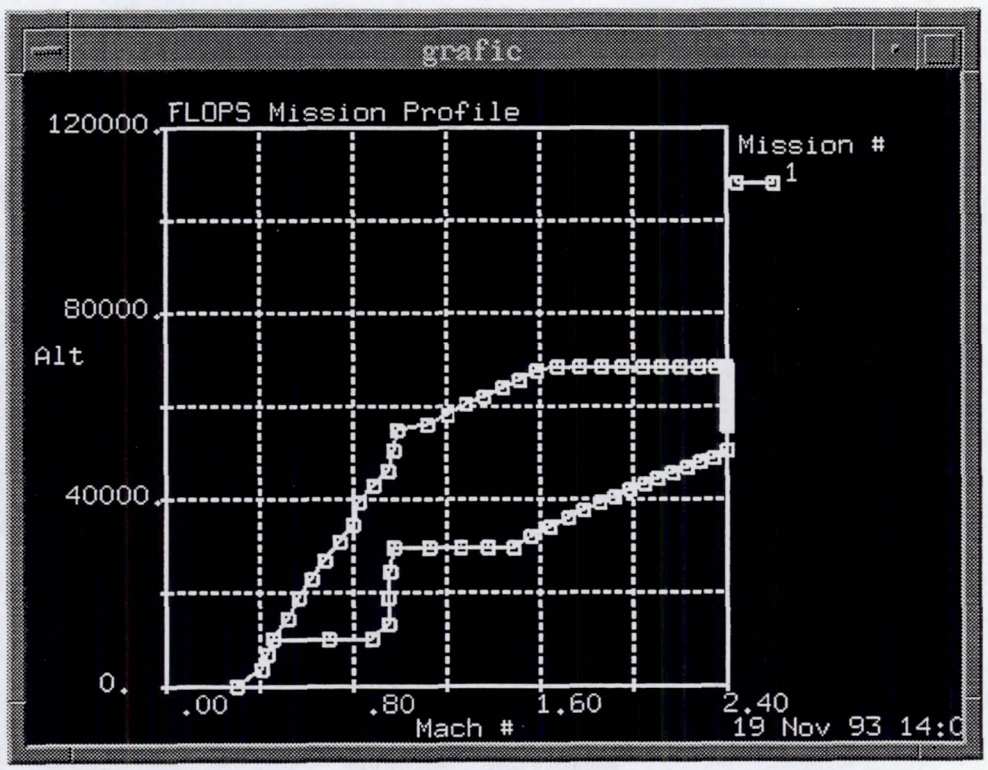


Figure 12: XFLOPS mission profile plot.

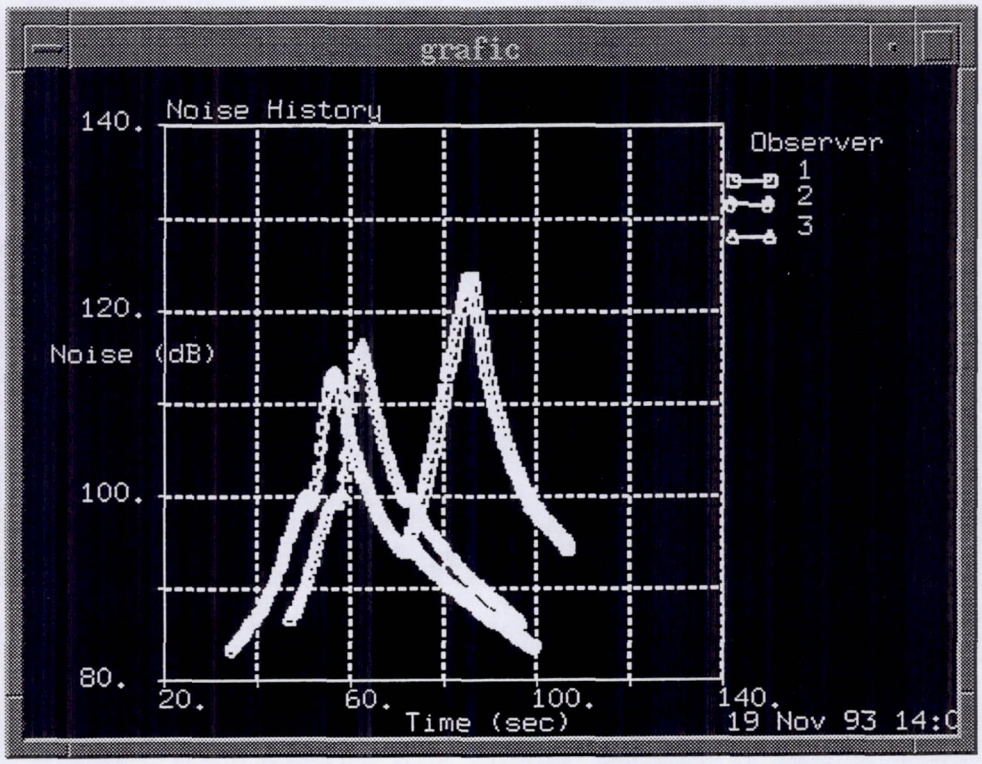


Figure 13: XFLOPS takeoff noise plot.

Contour Plots

The contour plotting is done with the XFLOPSPLOT plotting package. XFLOPSPLOT reads in a FLOPS thumbprint file, then displays forms for entering constraint and objective values. These forms display minimum and maximum values found in the contour file, simplifying data entry. When XFLOPSPLOT is executed, a contour plot is created. Figure 14 shows an XFLOPSPLOT constraint window, and figure 15 shows a sample thumbprint.

For more information on XFLOPSPLOT, see a copy of the on-line help in appendix C.

ON-LINE HELP

Both XFLOPS and XFLOPSPLOT use the NLEdit help facility. This feature allows users to access documentation for XFLOPS, XFLOPSPLOT, FLOPS, NLEdit, and Grafic. After a documentation file is chosen, a two-field window is created. The left field contains a table of contents for the help file. Users select topics by clicking on them with the mouse. The desired information is displayed in the right field. Figure 16 shows the help window displaying the FLOPS manual.

CONCLUSIONS

A graphical user interface, XFLOPS, has been created for the aircraft performance and sizing computer code FLOPS. This interface simplifies the use of FLOPS for both the experienced and novice users. Because the interface works both as a preprocessor and a post-processor, it augments the capabilities of FLOPS without interfering with its traditional use.

CONSTRAINTS

Description	Input	Min	Max
Approach Velocity	135.	130.801	143.267
Takeoff Field Length	11000.	8952.95	12318.8
Landing Field Length	11000.	8683.98	9918.82
Thrust Margin Missed Approach		-134584	2497.4
Thrust Margin Second Segment Climb Gradient		-142456	-7060.64
Excess Fuel Capacity		126098	256187
Compressor Discharge Temperature		0	0
Compressor Discharge Pressure		0	0
Jet Velocity		0	0
Specific Thrust		0	0
Aspect Ratio		0	0

Apply | Erase | Close | Help | Objec Gr | Range Gr

Figure 14: XFLOPSPLOT constraints window for contour plotting.

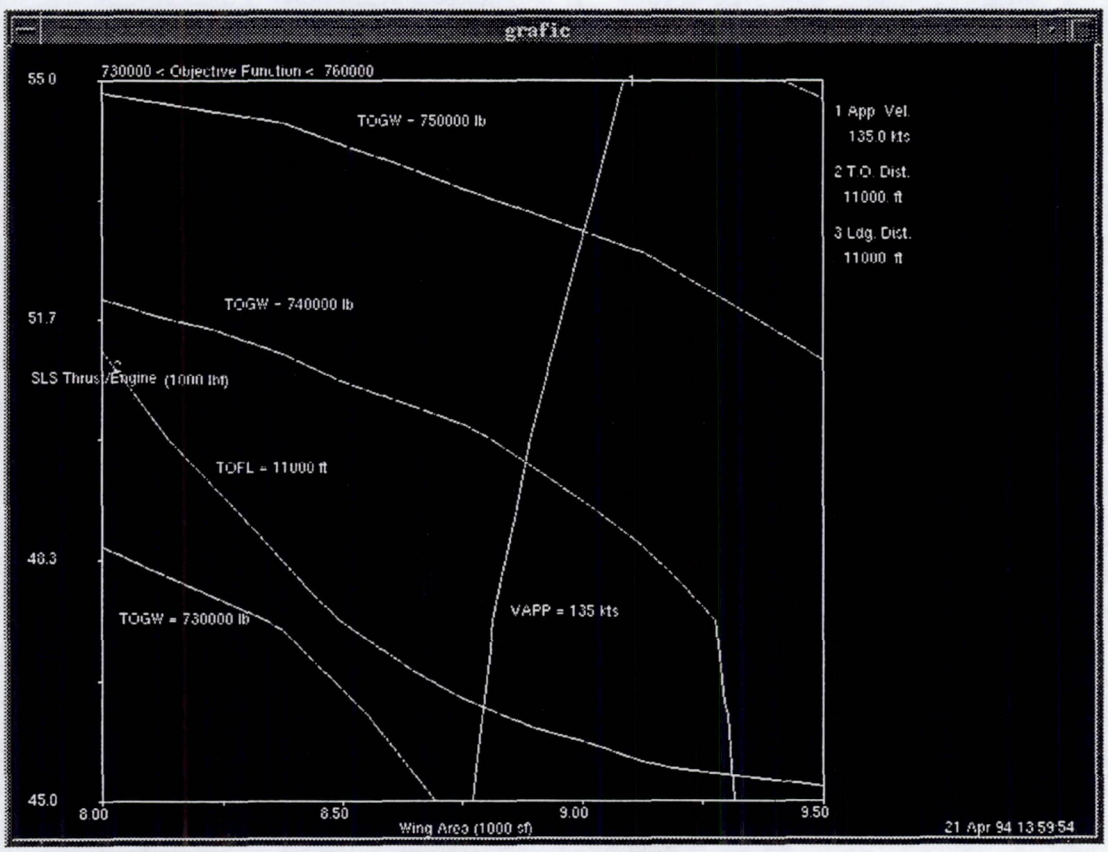


Figure 15: Sample thumbprint (takeoff weight as a function of engine and wing size).

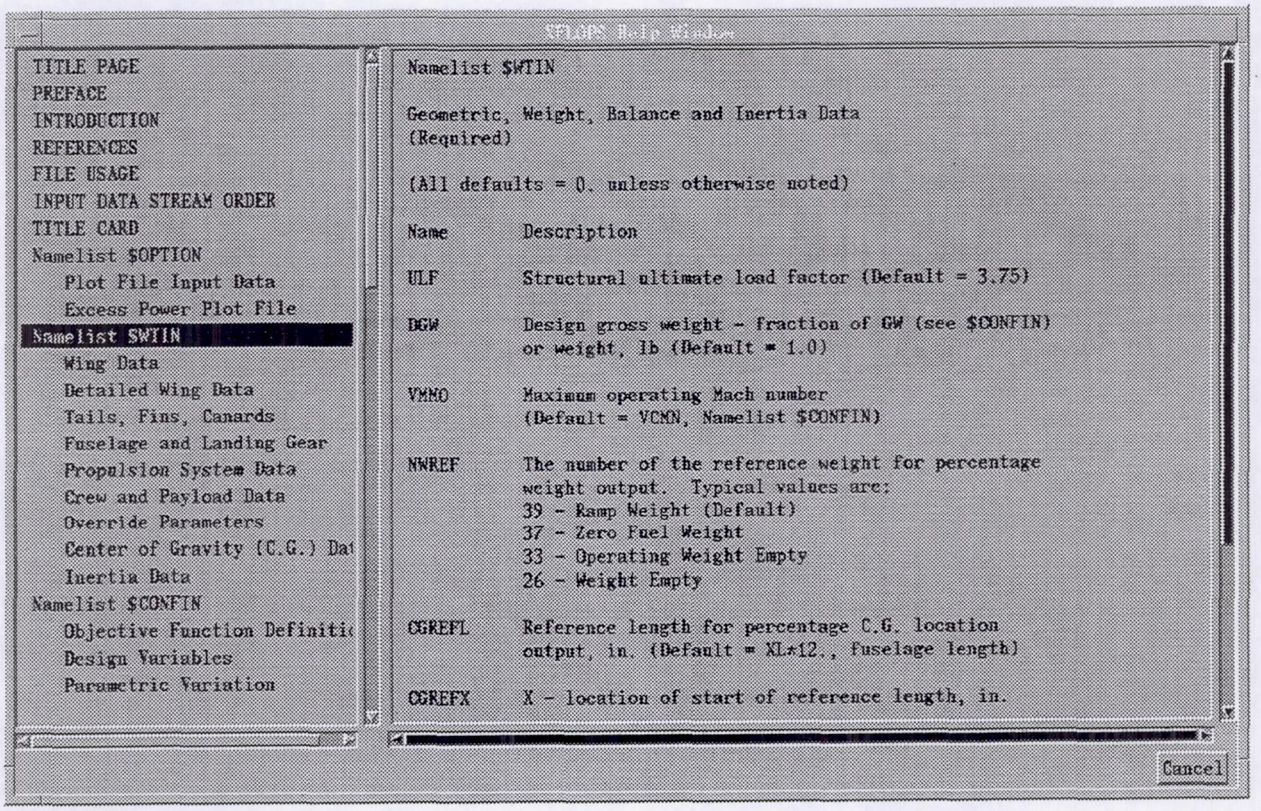


Figure 16: XFLOPS on-line help information (FLOPS manual).

The setup and execution of FLOPS are simplified by easy-to-use forms and editors in the interface. The task of analyzing the large quantity of data produced by FLOPS is simplified by an indexed output file and many plotting functions. In addition, user errors are decreased by improved organization of the input data, the extensive use of labeling forms, and the creation of on-line help information.

Based on the NLEdit graphical user interface framework, XFLOPS will be easy to maintain and modify as the analysis capabilities of FLOPS evolve. New parameters can be added to the XFLOPS interface without any graphics programming, and new plot types can be created with a minimal programming effort.

ACKNOWLEDGMENT

We thank Arnie McCullers, the creator of FLOPS, for his efforts in testing and debugging this software and for his suggested improvements.

APPENDIX A–FLOPS OVERVIEW

The Flight Optimization System (FLOPS) is a multidisciplinary computer program for conceptual and preliminary design and evaluation of advanced aircraft (ref. 1). It consists of nine primary modules: (1) weights, (2) aerodynamics, (3) engine cycle analysis, (4) propulsion data scaling and interpolation, (5) mission performance, (6) takeoff and landing, (7) noise footprint, (8) cost analysis, and (9) program control.

The weights module uses statistical and empirical equations to predict the weight of each item in a group weight statement. In addition, a more analytical wing weight estimation is available for use with more complex wing planforms. And centers of gravity and moments of inertia can be calculated for multiple fuel conditions.

The aerodynamics module uses a modified version of the EDET (Empirical Drag Estimation Technique) program to provide drag polars for performance calculations. Modifications include smoothing of the drag polars, more accurate Reynolds number calculations, and the inclusion of the Sommer and Short T' method for skin friction calculations. Alternatively, drag polars may be input and then scaled with variations in wing area and engine (nacelle) size.

The engine cycle analysis module is based on the QNEP (ref. 5) program, a modified version of NEPCOMP and its successors. It can internally generate an engine deck consisting of thrust and fuel flow data at a variety of Mach-altitude conditions. Engine cycle definition decks are provided for turbojets, turboprops, mixed flow turbofans, separate flow turbofans, and turbine bypass engines.

The propulsion data scaling and interpolation module uses an engine deck that has been input or one that has been generated by the engine cycle analysis module. The module fills in any missing data and uses linear or nonlinear scaling laws to scale the engine data to the desired thrust. It then provides any propulsion data requested by the mission performance module or the takeoff and landing module.

The mission performance module uses the calculated weights, aerodynamics, and propulsion system data to calculate performance. Optimum climb profiles based on energy considerations may be flown to start of cruise conditions. The cruise segments may be flown at the optimum altitude and/or Mach number for maximum range or endurance or to minimize NOx emissions, at the long-range cruise Mach number or at a constant lift coefficient. Descent may be flown at the optimum lift-drag ratio. In addition, acceleration, turn, refueling, payload release, and hold segments can be specified in any reasonable order. Reserve calculations can include flight to an alternate airport and a specified hold segment.

The takeoff and landing module computes the all-engine takeoff field length, the balanced field length (including one-engine-out takeoff and aborted takeoff), and the landing field length. The approach speed is also calculated, and the second segment climb gradient and the missed approach climb gradient criteria are evaluated. Insofar as possible with the available data, all FAR Part 25 requirements are met. The module also has the capability to generate a detailed takeoff and climbout profile to calculate noise footprints.

The noise footprint module uses the takeoff and climbout profile generated in the takeoff and landing module to compute noise footprint contour data or noise levels at user-specified or FAA locations. It is based on the FOOTPR (ref. 6) program. The noise sources include fan inlet and exhaust, jet, flap (for powered lift), core (combustor), turbine, and airframe. Noise propagation corrections are available for atmospheric attenuation, ground reflections, extra ground attenuation, and shielding.

The cost analysis (ref. 7) module uses configuration, engine, performance, and weights data from other modules. It can calculate airframe RDT&E cost, airframe production cost, engine RDT&E and production costs, direct operating cost, and indirect operating cost; and it can combine these costs to produce life cycle cost for subsonic transport aircraft.

Through the program control module, FLOPS can be used to analyze a point design, parametrically vary certain design variables, or optimize a configuration with respect to these design variables (for minimum gross weight, minimum fuel burned, maximum range, minimum cost, or minimum NOx emissions) by using nonlinear programming techniques. The configuration design variables are wing area, wing sweep, wing aspect ratio, wing taper ratio, wing thickness-chord ratio, gross weight, and thrust (size of engine). The performance design variables are cruise Mach number and maximum cruise altitude. The engine cycle design variables are the design point turbine entry temperature, the maximum turbine entry temperature, the fan pressure ratio, the overall pressure ratio, and the bypass ratio for turbofan and turbine bypass engines.

Most of the input data are in namelist format with default values coded into the program. For new users, we recommend that these default values be used whenever there is some uncertainty as to the meaning or appropriate value of a given parameter (i.e., "When in doubt, leave it out."). In most cases, this will provide reasonable results.

APPENDIX B–XFLOPS DOCUMENTATION

This appendix contains a dump of the XFLOPS on-line help information, covering all XFLOPS functions. The reader can get an idea of XFLOPS' capabilities by thumbing through this section.

The # characters in column one indicate section titles. They are used to by XFLOPS to create the table of contents.

```
#Introduction and Main Window

XFlops is an X-Windows/Motif graphical user interface for Flops.  It
allows you to read Flops input files, modify parameters, execute the
Flops batch code, and view/plot output data.

The main window is composed of a menu system, a message area, and a
matrix editing area for the configuration design variables.  The popup
menus allow the user to save and rename Flops input and output files,
restart and exit (File); examine and modify input data not contained in
a namelist (Input); examine and modify namelist input data (Namelists);
execute Flops and view the results (Execute); set namelist and graphical
display switches (Options); plot Flops output (Graphs); and view
instructions and manuals (Help).

#File Menu

The File Menu allows the user to save the Flops input and output files
created by the user interface.  It also enables the user to restart with
a new input file or exit.

#  Save FLOPSIN

The Save FLOPSIN Command will save the current values of the Flops
input variables into the file "FLOPSIN" or into the file name specified
if a new input file is being created.

#  Save FLOPSIN As...

The Save FLOPSIN As... Command will create a File Selection Box and
save the current values of the Flops input variables into the file
selected by the user.  "FLOPSIN" will not be affected.

#  New Input File...

This command starts the session over with a new input file.  It is
equivalent to exiting and restarting with a new file name.  A File
Selection Box is used to select the name of the new input file.

#  Change Output To...

The Change Output To... Command will create a File Selection Box and
change the internal name of the Flops output file from "FLOPOUT" to
the selected name.  The next time Flops is run, the output is put on
this file, otherwise, existing files are not affected.  If the name
```

of an existing output file is selected, "Display Output File" from the Execute Menu can be used to view that file.

Move Output To...

The Move Output To... Command will create a File Selection Box and move the "FLOPOUT" file created during "Run Flops" to the file selected by the user. "FLOPOUT" will disappear.

Exit

The Exit Command will end the XFlops session.

#Input Menu

The Input Menu gives the user the ability to look at and modify Flops input data not associated with a particular input namelist.

Title Card

The Title Card Command allows the user to input or edit the Flops title card which is the first line of the input data file. XFlops supports multiline titles, but Flops ignores all but the first line.

Engine Deck

The Engine Deck consists of one entry for each Mach-altitude-thrust combination. From 2 to 20 distinct Mach numbers may be input with 2 to 15 altitudes per Mach number. The number of altitudes per Mach number and the altitudes themselves do not have to be consistent between Mach numbers.

For NPCODE < 2, Power codes are ignored, and up to 16 thrust levels may be input for each Mach-altitude combination. Climb thrust must be input for each Mach-altitude point, and part power cruise data must be input for at least one Mach-altitude point. If IFILL > 0, part power data will be generated for any point for which part power data are not input.

For NPCODE > 1, thrust levels at each Mach-altitude combination are ordered using the power codes input. Only points with power codes corresponding to a PCODE in Namelist $ENGDIN will be used. If not all power code entries are input for a given Mach-altitude combination, the rest will be automatically generated, but a complete set - one point for each power code - must be input for at least one Mach-altitude combination.

If IDLE > 0, flight idle data will be generated for all points. The Engine Deck is order independent, and duplicate entries are ignored. The Engine Deck is terminated with a Mach number greater than 5. There is a limit of 1600 valid entries including duplicates and the terminator.

The engine data is displayed in the following array format:

Columns	Description
1	Mach number
2	Altitude, ft
3	Power code (used only if NPCODE > 1)
4	Gross thrust, lb
5	Ram drag, lb
6	Fuel flow, lb/hr
7	NOx Emissions Index, grams/kilogram fuel (or NOx Emissions, lb/hr or other engine parameter selected by user - See NOX, Namelist $ENGDIN)
8	Nozzle exit area, sq. in. (Required only if INSDRG = 1 in Namelist $ENGDIN)

The program uses the Net thrust = Gross thrust - Ram drag. Data appearing in unused columns are ignored.

Engine data may be modified using this menu pick only if it is part of the input deck (IGENEN = 0 in Namelist $ENGDIN). If the engine data is from an external file (IGENEN = -1), it can be modified with the Editor. If IGENEN =1, the engine deck will be generated during Flops execution and this menu pick has no function.

For more information on the engine deck, see the Flops Manual under the Help Menu.

Mission Definition

Climb, cruise, and descent schedules are defined in Namelist $MISSIN along with a complete definition of the reserve mission. These schedules are combined with other segments to define the main mission. The segments available are defined below. They can be sequenced in almost any logical order. The deck must start with a START card and end with an END card. The first segment will be a CLIMB or an ACCEL card and the last will be a DESCENT card. All CLIMB segments should be followed by a CRUISE or HOLD segment, and DESCENT segments should be preceded by a CRUISE or HOLD segment. Continuity in Mach number, altitude and energy is maintained wherever possible.

If Namelist $RERUN is used, there will be multiple occurrences of the Mission Definition Data. The set of data that appears is slaved to the current occurrence number for Namelist $MISSIN. To edit a different set of Mission Data, change the Namelist $MISSIN number.

The mission data are entered in the array format shown below. Any undescribed columns are not used for that segment, however, for the Xbae Matrix Editor to work properly, each cell must contain something. The default value initially put into each cell is a single blank space.

Segment Definition Cards (Maximum of 40)

Columns	Contents
1	START
4	Starting Mach number (Default = CLMMIN for first climb segment)
5	Starting altitude, ft (Default = CLAMIN for first climb segment)
1	CLIMB
3	Climb schedule number (Default = 1)
6	Starting Mach number (Default = ending Mach number for previous segment or START card)
7	Starting altitude, ft (Required if starting Mach number is specified)
1	CRUISE
3	Cruise schedule number (Default = 1)
4	Total distance at the end of this cruise segment, n.mi. Required for all but the "free" cruise segment (see notes below)
1	REFUEL
4	Fuel added, lb
5	Time required, min
1	RELEASE
4	Weight of payload released, lb
1	ACCEL
2	Number of store drag condition to be applied to this segment
3	Acceleration power setting (Default = 1, max power)
4	Ending Mach number
6	Starting Mach number (Default = ending Mach number for previous segment) If this is input, distance is not included in mission range.
7	Altitude, ft (Default = ending altitude for previous segment)
8	Drag coefficient increment
9	Weight specified for point performance
1	TURN (Sustained turn solution is for instantaneous condition, i.e., weight at start of turn. For long turns, use multiple turns.)
2	Number of store drag condition to be applied to this segment
3	Power setting for max G turn (Default is specified G turn)
4	Turn arc, deg
5	Turn acceleration, G's (Not used if power setting is specified)
6	Starting Mach number (Default = ending Mach number

```
            for previous segment)
   7        Altitude, ft (Default = ending altitude for
            previous segment)
   8        Drag coefficient increment
   9        Weight specified for point performance

   1        HOLD
   3        Cruise schedule number (Default = 1)
   4        Holding time, min

   1        DESCENT

   1        END
   4        Ending Mach number (Default = DEMMIN)
   5        Ending altitude (Default = DEAMIN)
```

Notes on the "free" segment. Flops flies missions from both ends meeting at the "free" CRUISE segment. It starts with the ramp weight, subtracts taxi out and takeoff allowances, and then climbs, cruises, etc. until the free segment is reached. It then starts with the zero fuel weight, adds the reserve fuel, and then descends, cruises, etc. in reverse until the end of the free segment is reached. The difference in fuel between the start and end of the free segment is then used to determine the free segment cruise distance. The free segment is usually the highest energy (altitude) segment. CRUISE segments after the free segment must have a negative distance indicated, the distance from the end of the mission to the start of that segment. CLIMB segments may not follow the free segment, and DESCENT segments may not precede the free segment, except for military (instantaneous) descents which are not defined as segments except for the final descent.

Aero Data - CDi

The Aero Data - CDi Command allows the user to examine and modify the CDi aero data. This data is presented in an array format. The top left hand location is blank. The values in the top row are the Mach numbers in increasing order. The values in the left column are CL values in increasing order. The rest of the values are CDi as a function of CL and Mach number. If there is no CDi data in the input file (MYAERO = 0 in Namelist $AERIN) then this menu pick will do nothing.

For more information, see the Flops Manual under the Help Menu.

Aero Data - CDo

The Aero Data - CDo Command allows the user to examine and modify the CDo aero data. This data is presented in an array format. The top left hand location is blank. The values in the top row are the altitudes in increasing order. The values in the left column are Mach numbers in increasing order. The rest of the values are CDo as a function of altitude and Mach number. If there is no CDo data in the input file (MYAERO = 0 or 2 in Namelist $AERIN) then this menu pick will do nothing.

For more information, see the Flops Manual under the Help Menu.

Aero Data - CDwave

The Aero Data - CDwave Command allows the user to examine and modify wave drag data if it is input as formatted data (IWAVE = 1 in Namelist $AERIN). This data is presented in an array format. The top row is the Mach numbers in increasing order and the bottom row is the corresponding wave drag coefficients.

For more information, see the Flops Manual under the Help Menu.

#Namelists

The Namelists Command allows the user to modify inputs in a particular namelist field. This command will present a list of the different Flops namelists. The user then must pick the particular namelist to edit. When a namelist is chosen, a NLEdit window is created to modify the inputs. If the namelist can have multiple occurrences, values for the last one input or accessed will be shown.

This window consists of input fields for each namelist variable and comment lines. These comments will be included in NLEdit's namelist output file preceding the corresponding namelist.

There are four different types of input fields for the namelist variables; toggle fields, multiple choice fields, filename fields, and text fields. Toggle fields are used for variables that can have one of two values (usually 1 or 0, true or false). Multiple choice fields are used for variables that can have one of several values. Filename fields are used for file name inputs, and text fields are used for all other types of input.

Each data field has a push button variable name, an input field, and a one-line description. Clicking on the variable name will display more detailed help information about that variable. Pressing the F1 function key after selecting a field will display a window containing all of the information about that variable.

Using toggle and multiple choice fields is straight-forward. Screen highlighting is used to give the buttons a three dimensional appearance; in is on or true or 1 and out is off or false or 0. Multiple choice buttons have a "radio box" behavior, that is, only one button can be pressed in at a time.

The filename and text input fields work the same. The user activates the input field and types in the data. File names (including the directory path, if necessary) are enclosed in single quotes (e.g., '../Edata/engine.new'). All other inputs are handled as text inputs, i.e., they are read in and stored as character arrays. This applies to integer, real, and double precision numerical values as well as equations. For integers, up and down arrow buttons are provided to increment and decrement the value, respectively. Simple equations involving predefined variables, logarithmic and trigonometric functions,

and scientific notation may also be used in the text field to define variable values.

If the namelist can have multiple occurrences, additional options are provided at the bottom of the window. The box shows the occurrence number which can be changed with the up and down arrows. Also, an occurrence of this namelist with default values may be Added, a Copy of the current occurrence may be added, the current occurrence may be Deleted, or the values in the current occurrence may be Reset to default values.

For more information, see the NLEdit Manual under the Help Menu.

For more information on the individual namelists, see the Flops Manual under the Help Menu.

Additional Capabilities

There are additional capabilities accessible via a popup menu. This menu is displayed by pressing the right mouse button while the mouse pointer is in a text field. The following options are available depending on the variable type.

Help - popup help window. Equivalent to clicking on the variable name.

Edit - popup a larger editing window. Good to use if the regular input window is too small.

Array Edit - popup an array editor. Displays a row-column input field for editing arrays.

File Selection Box - popup a file selection box. Displays a list of possible input files. Used for filename inputs.

Edit Contents - popup a file editor. Displays an editor containing the selected input file. Used for filename inputs.

Show Calc - display numerical value. Displays the numerical value of a mathematical function.

Show Default - display default value.

Show Min - display minimum value.

Show Max - display maximum value.

Refresh - display actual value.

For more information, see the NLEdit Manual under the Help Menu.

Error Checking

The user interface also provide built-in error checking. Errors are caused by a type mismatch, value outside range, or an equation that the

calculator routine cannot calculate. Normally, if an error occurs, a warning will be displayed in the message window and the text field will be displayed in reverse video. The user may correct the data or ignore the message and proceed. Inputs are checked for errors when the contents of the text fields are entered into the data dictionary. This occurs when the Enter key or Apply button is pressed.

For more information, see the NLEdit Manual under the Help Menu.

Multiple Occurrences

Some namelists may occur in a file more than once. The text field just above the Apply button indicates the occurrence number of the namelist. For example, if the occurrence number is 2 then the values shown in the fields will be printed the second time the namelist occurs in the output file. Buttons are provided just after the occurrence number text field to insert a new namelist occurrence, copy current values of the namelist to a new occurrence, delete an occurrence of the namelist, and to reset all the values in the current occurrence to their default values.

For more information, see the NLEdit Manual under the Help Menu.

#Execute

The Execute menu provides the ability to run Flops in either batch or design mode, view the output, search for variables in the data base, and edit files.

Run Flops

The Run Flops Command will save the current values of the Flops input variables into the file "FLOPSIN" and then execute Flops using this input file, creating the output file "FLOPOUT".

Search...

the Search Command enables grouping of variables related to a particular item or concept. When activated, a window is created with an input field for a string and toggle buttons to search the variable names, label fields, and help fields for the string. The search command will search the variable name fields, label fields, and help fields of all the database variables for matches to this string. A list of variables that contain matches is then displayed. The user is then given the option of creating an input form from these variables. This form will contain all the matching variables in the input window.

For example, suppose the user wished to locate all the variables that pertained to aircraft noise. The user could search the Flops database for all variables with information that matched the string "NOISE". XFlops would return a list of all the variables pertaining to noise. The user could then create an input form for these variables, giving him a window with all the Flops variables pertaining to noise in one place regardless of their original namelists.

Display Output File

This command provides a convenient way to view the Flops output file. FLOPS output is set up with special characters to provide a table of contents and divide the data into meaningful, manageable chunks.

Design Mode

The Design Mode menu pick creates a screen containing current values of the design variables, the constituents of the objective function, and the constraints. The user may change the design variable values and run Flops to see the effect of the changes, or select certain variables to be active and run an optimization drawdown. Expert advice is available for each constraint by clicking on the "OK" or "Violated".

Editor

The Editor Command creates an editing window that can be used to look at and modify any file. From the Editor "File" menu, the user may Open a file, Save the file after modifying it, Save As another file, Clear the editing window, or Close the window (exit). These files do not have to be Flops related. The Editor's "Edit" menu contains typical editing functions such as Cut, Copy, Paste, Delete and Search.

In addition, the "Display" menu in the Editor has several selections hard wired to several Flops files. The XFlops input, XFlops output (FLOPSIN), Flops output (FLOPOUT), plotx input (xflops.plotx), and external engine deck (ENGDEK) files can all be viewed from a menu pick under the Display option; eliminating the need to find and input these file names.

#Options

The Options Menu gives the user the ability to modify certain aspects of the way XFlops displays Namelists and plots.

Show Defaults

The Show Defaults Toggle Switch determines if the default values are to be shown. These default values are specified in the xflops.hlp setup file.

If on, the default values will be shown in any input field not containing data.

If off, a blank input field will be shown in any input field not containing data.

Show Only Used Parameters

The Show Only Used Parameters Toggle Switch determines which variables will be displayed.

If on, only variables contained in the input file will be displayed.

If off, all the Flops variables are displayed.

This eliminates the need to display all the Flops input variables when the only ones of interest are those contained in the current input file. Messages are always displayed as well as Boolean (On/Off) variables whose default value is On.

Test Limits

The Test Limits Toggle Switch determines if the limit tester is on. The limit tester checks all input values against minimums and maximums defined in the xflops.hlp setup file.

If on, all input values are checked. If a value is out of bounds, a warning message is displayed in the message window and the input field is displayed in reverse video. The user may correct the data or ignore the message and proceed.

If off, input values are not checked.

Run Plotx in Foreground

The Run Plotx in Foreground determines whether plotx is run in the foreground or background when plots are created.

If on, plotx is run in the foreground. The advantage is that in this mode plotx will take command of the operating window. This means that it is possible to enter ASCII commands into the plotting package from the operating window (certain Plotx functions require ASCII inputs). The disadvantage is that the plot must be exited before any other XFlops commands can be given.

If off, plotx is run in the background. The advantage is that XFlops is in control of the operating window. This means that other XFlops functions, including the creation of other plots, can be executed while this plot is still showing.

For more information on plotx, see the Plotx Manual under the Help Menu.

White Background for Plots

This option toggles between plots with white lines on a black background and black lines on a white background. The default is white lines on a black background unless the environment variable "GRAFIC_BACKGROUND" has been set to 'white'.

#Graphs Menu

The Graphs Menu gives the various options for plotting different things pertaining to the Flops output. Please note that these functions only require that the files that they use exist. Thus, it is not necessary to run Flops through XFlops to take advantage of the plotting functions. Output data can be plotted from any set of output files with the correct names. This can create problems, however, in that if the latest Flops

run fails to create the proper file, an old one may be plotted instead.

Help file descriptions for most of the plot options indicate the type of plot generated, options as to how that data may be plotted, the file required, where its name is input in the Flops input file, the default name, and the Flops input requirements to generate that file or the required data.

Special LeRC Options

Available only at Lewis Research Center.

Help for Plot Commands

The GRAFIC plotting package used in XFlops has the capability for on-screen editing of plots. This menu pick displays a user's guide for these commands in logical groups. This capability includes changing the plot size, adding lines and annotation, saving and printing plots, picking off data, and other functions.

Note - Plot commands requiring input from the calling window will function as described only if the "Run plotx in foreground" option is turned on in the "Options" Menu from the main screen.

Fuselage Drawing

The fuselage drawing option provides a capability to view the results of a detailed fuselage design performed by Flops if the fuselage length is not input. See the Flops manual, Namelist WTIN, Crew and Payload data.

This option draws a representative schematic of the fuselage interior layout. The layout shown is not the only one possible and should be used only as an indicator of what the passenger compartment would look like.

Fuselage data is passed from Flops to XFlops in a file called FUSEDATA. This file may be plotted even if it was not generated in the current session.

Contour (Thumbprint) Plot

Plot: Contour or Thumbprint Data with Constraints.

From File: CNFILE (Namelist $OPTION, Default = 'FLPPLOT')

Requirements: IOPT = 4 in Namelist $OPTION

The Contour Plot Command activates the XFlopsPlot plotting package which was designed for creating thumbprint plots. It is possible, however, to select any contour plot file once in the XFlopsPlot plotting package. For more information on XFlopsplot, see the help available with XFlopsPlot.

Note: Although Flops is set up to plot any two variables, XFlopsPlot can only handle wing area and thrust. Hopefully, this will be fixed

soon.

Quick Contour Plot

The Quick Contour Plot Command activates the XFlopsPlot plotting package without going through the XFlopsPlot interface. The values that were used for constraints and contour functions from the last saved setup are used to display a thumbprint.

The advantage to this is that if the setup for the current desired thumbprint is the same as the last saved setup, it is not necessary to go through the trouble of generating this data.

For more information on XFlopsPlot and the Save Setup feature see the help available with XFlopsPlot.

Mission Summary Plots

Plot: Detailed Mission Summary Data. The user can choose the X and Y coordinates from a list of 27 possible parameters.

From File: MSFILE (Namelist $OPTION, Default = 'MSMPLOT')

Requirements: IXFL = 1 in Namelist $OPTION

Takeoff Profile Plots

Plot: Detailed Takeoff and Climbout Data. The user can choose the X and Y coordinates from a list of 14 possible parameters.

From File: NOFILE (Namelist $OPTION, Default = 'NPROF')

Requirements: NOPRO = 1 and NPFILE = 1 in Namelist $OPTION

Cruise Schedule Plots

Plot: Cruise Summary Data for all cruise schedules from Ramp Weight to Zero Fuel Weight. The user can choose the X and Y coordinates from a list of 16 possible parameters.

From File: CRFILE (Namelist $OPTION, Default = 'CRUPLOT')

Requirements: IXFL = 1 in Namelist $OPTION
IFLAG > 0 in Namelist $MISSIN

Ps or Nz Plots

Plot: Excess Power (Ps) or Load Factor (Nz) Data. The user can choose which of the plots that he has defined in Namelist $OPTION is to be plotted, excess power (instantaneous rate of climb capability) contours at specified load factors or load factor contours with specified minimum rates of climb. He may select the contours to be plotted or plot default contours calculated internally.

From File: PSFILE (Namelist $OPTION, Default = 'PSPLOT')

Requirements: IPLTPS > 0 in Namelist $OPTION

Edit Plotx Data

This command allows the user to modify graphs created by XFlops. It gives the user the ability to modify the graph size, line colors, titles, and axes.

To use this option the user must have created a plot. When a plot is created: 1) inputs for the desired graph are stored in the plotx section of the database, 2) these inputs are then dumped to a plotx input file, and 3) plotx is then executed to create the graph.

Since the Plotx inputs are stored in the database, it is possible for the user to modify them. The Edit Plotx Data Command creates an input window for editing Plotx data. After changing the inputs, the new graph can be seen using the Plot Plotx Data command.

For more information on the Plotx plotting package, see the Plotx Manual under the Help Menu.

Plot Plotx Data

The Plot Plotx Data Command is used in conjunction with the Edit Plotx Data Command above to modify existing graphs. Once the data in the namelist has been modified, this command will create the new plot. Editing the existing "xflops.plotx" file will not affect the plot since a new "xflops.plotx" file will be generated from namelist data in the data base when this command is executed.

For more information on the Plotx plotting package, see the Plotx Manual under the Help Menu.

Aero Polar Plots

Plot: Aerodynamic Polars. The user can choose the type of plot from 7 options - CD or CDi vs. CL or CL vs. L/D for each Mach number, Mach number vs. CD, CDi or L/D for each CL, or CDwave and CDo vs Mach number. The user must also select which set of polars is to be plotted (from 1 to N) since there may be more than one set on the file due to Parametric Variation, Rerun, etc.

From File: APFILE (Namelist $OPTION, Default = 'POLPLOT')

Requirements: IPOLP > 0 in Namelist $OPTION

Engine Plots

Plot: Engine Performance Data. The user can choose the type of plot from 14 options involving thrust, fuel flow, SFC, NOx Index (or other engine parameter), Mach number, and

```
                altitude.  The user must also select the Mach number,
                altitude, or power code for the plot depending on the
                type of plot.

From File:      THFILE (Namelist $OPTION, Default = 'THRPLOT')

Requirements:   IPLTTH > 0 in Namelist $OPTION

#  Noise Plots

Plot:           Noise Data.  The user can choose the type of plot - noise
                histories or distance histories.

From File:      FLOPOUT - the Flops printed output file

Requirements:   NOISE = 1 in Namelist $OPTION
                NDPRT = 1 in Namelist $NOISIN for noise histories
                IGEOM = 1 in Namelist $NOISIN for distance histories

#  Climb Plots

Plot:           All Climb Schedule Data printed by Flops during mission
                analysis.  The user can choose the type of plot -
                altitude vs. distance,  altitude vs. Mach number,
                time vs. Mach number,   L/D vs. Mach number,
                Cl vs. Mach number,     throttle vs. Mach number,
                thrust vs. Mach number, q vs. Mach number,
                weight vs. Mach number, ROC vs. Mach number.

From File:      FLOPOUT - the Flops printed output file
                Note - files named .climb1, etc. are created

Requirements:   IFLAG > 1 in Namelist $MISSIN

#  Descent Plots

Plot:           All Descent Schedule Data printed by Flops during mission
                analysis.  The user can choose the type of plot -
                altitude vs. distance,  altitude vs. Mach number,
                time vs. Mach number,   L/D vs. Mach number,
                Cl vs. Mach number,     throttle vs. Mach number,
                thrust vs. Mach number, q vs. Mach number,
                weight vs. Mach number, ROC vs. Mach number.

From File:      FLOPOUT - the Flops printed output file

Requirements:   IFLAG > 1 in Namelist $MISSIN

#Help Menu

The main Help Menu provides information about and instructions on how to
use the XFlops system.  In addition, user documentation for some of the
main components of the system are presented in an indexed, easy to read
format.
```

About XFlops...

This provides names and addresses of the primary authors of XFlops and Flops as well as very brief descriptions.

XFlops Manual...

This command provides instructions on how to use the XFlops menu system and descriptions of some of the system's capabilities.

Flops Manual...

This command provides a convenient way to view the current version of the Flops User's Guide.

NLEdit Manual...

This provides more information on the NLEdit Generic Graphical User Interface used in XFlops.

Plotx Manual..

This provides more information on the Plotx line plotting capability. It also includes a description of the Plotx Namelist &DATA variables.

Grafic Manual...

This provides more information on the Grafic library of interactive graphics routines used in XFlops.

#Additional Documentation

There are two PostScript files in directory Xbae/doc which, when routed to a printer, will provide documentation on XbaeCaption and XbaeMatrix (10 and 31 pages, respectively).

There is also a PostScript file, containing the NLEdit manual complete with figures, in the tarred and compressed directory Doc.tar.Z in directory nledit.

APPENDIX C–XFLOPSPLOT DOCUMENTATION

This appendix contains a dump of the XFLOPSPLOT on-line help information, covering all XFLOPSPLOT functions. The reader can get an idea of XFLOPSPLOT's capabilities by thumbing through this section.

The # characters in column one indicate section titles. They are used to by XFLOPSPLOT to create the table of contents.

```
#Introduction

XFlopsPlot is an X-Windows/Motif graphical user interface for
FlopsPlot.  It will allow you to read in contour plot files, set
constraints, and display contour plots.  XFlopsPlot can be either
used in conjunction with XFlops or as a stand alone code.

There are two methods to run XFlopsPlot.  The first method uses
the graphical user interface.  This option is invoked using the
following command line statement:

                    xflopsplot filename

where "filename" is the name of a contour plot data file in the
same format as generated by Flops.

The second method directly creates a thumbprint using the data in
the contour file augmented by setup data in the ".plottmp" file
previously created using the Save Setup command.  This method
should be used if no setup data is desired (if an old ".plottmp"
file exists, it must be deleted or renamed) or if the data in the
current ".plottmp" setup file is what the user wants.  This option
is invoked by adding a "q" to the previous command line statement:

                    xflopsplot filename q

This command is equivalent to the Quick Contour plot option in XFlops.

#File

The file menu contains several options relating to file management.

#  Open File

The Open File command displays a window allowing the user to choose
different thumbprint input files.

#  Edit File

The Edit File command opens an editing window allowing the user to
examine, edit, and save files.  When this window is first opened, it
displays the contents of the XFlopsPlot input file.  However, any file
may be examined with this editor.

#  Save Setup
```

The Save Setup command saves the contour values, constraint values, and type of graph in a temporary file. This file is automatically read in when XFlopsPlot is started. This method eliminates the need to re-enter constraint values when they are always the same for different thumbprints. Additionally, the Save Setup command can be used to created thumbprints without directly running XFlopsPlot (see Introduction).

Exit

Exit XFlopsPlot.

#Input

The input menu pick gives the user the ability to input contour and constraint values.

Objective Contours

This window controls the objective contour values that are plotted. To plot specific contours, enter their values into the text input window and press the Graph w Inputs button. To plot the default contours, press the Graph w Defaults button. If any of the input contours are out of the thumbprint range, the text input video is reversed and a warning message is displayed. This is just a warning and can be ignored.

Initially, the contour values are set to the values from the last saved setup, if a setup file exists.

Range Contours

This window controls the range contour values that are plotted. To plot specific contours, enter their values into the text input window and press the Graph w Inputs button. To plot the default contours, press the Graph w Defaults button. If any of the input contours are out of the thumbprint range, the text input video is reversed and a warning message is displayed. This is just a warning and can be ignored.

Initially, the contour values are set to the values from the last saved setup, if a setup file exists.

Constraints

This window contains the constraint values that are plotted. Up to 5 values may be input for each constraint. If any of the constraint values are out of the thumbprint range, the text input video is reversed and a warning message is displayed. This is just a warning and can be ignored.

The Apply button is equivalent to pressing return after each input. The Erase button empties all the input fields. The Object Gr and Range Gr buttons create objective and range graphs with the format specified in the objective and range windows (this eliminates the need

to re-enter the objective or range window when changing a constraint).

Initially, the constraint values are set to the values from the last saved setup, if the setup file exists.

WARNING: The way XFlopsPlot is currently set up, the constraints must have data in the minimum and maximum columns or the window will be made incorrectly. This means that a contour file MUST be read in before the constraint window is opened.

#Options

This window allows the user to change the appearance of the contour plot. The first four options specify the type of contours to be plotted. When using the 220 color rainbow palette, particularly with Color Constraint Lines, it is possible to exceed the maximum number of definable colors (usually 256), indicated by messages like "can't allocate color". This generally occurs when calling XFlopsPlot from XFlops after modifying a Namelist or running the Design Mode option. To get rid of the problem, run XFlopsPlot by itself when using the 220 color palette.

Mono Contour Lines

Turning on this option will result in discrete monochromatic contour lines - white on black or black on white depending on the "White Background for Plots" option below. This option can be used with the "Continuous Color Shading" option.

Color Contour Lines

This option will result in discrete contour lines using the 220 color rainbow palette. This option cannot be used with any other contour plot option.

Color Contour Bands

This option causes the objective range to be divided into 13 equal bands. Each band is filled with a color according to a 13 color rainbow palette. This option cannot be used with any other contour plot option.

Continuous Color Shading

This option shows the contours as continuously changing colors using a 220 color rainbow palette. It can be used with the "Mono Contour Lines" option to superimpose contour lines on the color shading.

Color Constraint Lines

This option causes the constraint lines and legends to be color coded. It can be used with any other option.

White Background for Plots

This option toggles between plots with white lines on a black background

and black lines on a white background. The default is white lines on a black background unless the environment variable "GRAFIC_BACKGROUND" has been set to 'white'. If XFlopsPlot is called from XFlops, the value in XFlops is used as the default.

REFERENCES

1. McCullers, L.A.: Aircraft Configuration Optimization Including Optimized Flight Profiles, Multidisciplinary Analysis and Optimization Part I. NASA CP-2327, 1984.
2. Curlett, Brian P.: NLEdit–A Generic Graphical User Interface for FORTRAN Programs. NASA TM-4543, 1993.
3. Young, D.A.: The X Windows Programming and Applications With Xt–OSF/Motif Edition. Prentice-Hall, Englewood Cliffs, New Jersey, 1990.
4. Open Software Foundation: OSF/Motif Programmer's Reference. Prentice-Hall, Englewood Cliffs, New Jersey, 1991.
5. Geiselhart, Karl A.: A Technique for Integrating Engine Cycle and Aircraft Configuration Optimization. NASA CR-191602, 1994.
6. Clark, Bruce J.: Computer Program to Predict Aircraft Noise. NASA TP-1913, 1981.
7. Johnson, V.S.: Life Cycle Cost in the Conceptual Design of Subsonic Commercial Aircraft. Ph. D. Dissertation, Univ. of Kansas, Lawrence, Kansas, May 1989.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 1994	Technical Memorandum

4. TITLE AND SUBTITLE

Graphical User Interface for the NASA FLOPS Aircraft Performance and Sizing Code

5. FUNDING NUMBERS

WU–505–69–50

6. AUTHOR(S)

Thomas M. Lavelle and Brian P. Curlett

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135–3191

8. PERFORMING ORGANIZATION REPORT NUMBER

E–8957

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, D.C. 20546–0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TM–106649

11. SUPPLEMENTARY NOTES

Responsible person, Thomas M. Lavelle, organization code 2410, (216) 433–7042.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 05

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

XFLOPS is an X-Windows/Motif graphical user interface for the aircraft performance and sizing code FLOPS. This new interface simplifies entering data and analyzing results, thereby reducing analysis time and errors. Data entry is simpler because input windows are used for each of the FLOPS namelists. These windows contain fields to input the variable's values along with help information describing the variable's function. Analyzing results is simpler because output data are displayed rapidly. This is accomplished in two ways. First, because the output file has been indexed, users can view particular sections with the click of a mouse button. Second, because menu picks have been created, users can plot engine and aircraft performance data. In addition, XFLOPS has a built-in help system and complete on-line documentation for FLOPS.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Aircraft design; Interactive graphics	39
	16. PRICE CODE: A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	